DEFENCE RESEARCH ESTABLISHMENT
Division of Command and Control Warfare Technology
P O Box 1165
SE-581 11 LINKÖPING SWEDEN

FOA-R--00-01496-408--SE
April 2000
ISSN 1104-9154

Thomas Kaijser

The Primal-Dual Algorithm for the Assignment Problem

Distribution: Celsius Tech, EMW, FMV
FOA: 1, 2, 3, 35, 6, 7, 77

Issuing organization	**Document ref. No., ISRN**	
Defence Research Establishment Division of Command and Control Warfare Technology P O Box 1165 SE-581 11 LINKÖPING SWEDEN	FOA-R--00-01496-408--SE	
	Date of issue April 2000	**Project No.** E3028
	Project name (abbrev. if necessary)	
Author(s)	**Initiator or sponsoring organization**	
Thomas Kaijser	**Project manager** Hans Hellsten	
	Scientifically and technically responsible Thomas Kaijser	

Document title

The Primal-Dual Algorithm for the Assignment Problem

Abstract

The assignment problem means to find the best way to connect, in a one-to-one manner, the objects of one class of objects to another class of objects. For each pair of objects there is a predefined value – called the cost, and the best total connection is obtained when the total cost for alla pairs is as small as possible.

This is a classical problem in optimization theory with many applications, for example to logistics and image analysis.

The purpose of this paper is to present the classical primal-dual algorithm in a transparent and selfcontained way. The presentation does not rely on concepts or basic results from the theory of optimization.

Keywords

Assignment problem, transportation problem, primal-dual algorithm

Further bibliographic information	**Language** English	
ISSN 1104-9154	**ISBN**	
	Pages 23 p.	**Price** Acc. to pricelist

Distributor (if not issuing organization)

Dokumentets utgivare	**Dokumentbeteckning, ISRN**	
Försvarets forskningsanstalt Avdelningen för Ledningssystemteknik Box 1165 581 11 LINKÖPING	FOA-R--00-01496-408--SE	
	Dokumentets datum April 2000	**Uppdragsnummer** E3028
	Projektnamn (ev förkortat)	
Upphovsman(män) Thomas Kaijser	**Uppdragsgivare**	
	Projektansvarig Hans Hellsten	
	Fackansvarig Thomas Kaijser	

Dokumentets titel i översättning

Primaldualalgoritmen för tillordningsproblemet

Sammanfattning

Med tillordningsproblemet avses att finna bästa sätt att förbinda, på ett parvist sätt, elementen i en mängd med elementen i en annan mängd. För varje par av element från de två mängderna finns givet ett förbestämt tal, som kallas kostnaden, och med den bästa förbindelsen menas den totala parbildning vars totala kostnad är minst.

Tillordningsproblemet är ett klassiskt problem i optimeringslära med många tillämpningar, t ex inom logistik och bildanalys.

Syftet med denna uppsats är att beskriva primaldualalgoritmen för lösandet av tillordningsproblemet på ett tydligt sätt. Framställningen är matematisk, men vilar inte på begrepp eller teori från optimeringslära.

Nyckelord

Tillordningsproblemet, transportproblemet, primaldualalgoritmen

Övriga bibliografiska uppgifter	**Språk** Engelska	
ISSN 1104-9154	**ISBN**	
	Omfång 23 s.	**Pris** Enligt prislista

Distributör (om annan än ovan) V.1.1

Contents.

1 Introduction. The assignment problem.

The purpose of this paper is to present the primal-dual algorithm for the assignment problem in a transperent way, for readers who are not trained in optimization theory but do have som basic mathematical knowledge.

The assignment problem can be described as follows. Let $\{R_i, i = 1, 2, ..., N\}$ be a set of N points which, for sake of thought, we assume has its elements located in the plane. Each point R_i will be called a source. Let $\{S_j, j = 1, 2, ..., M\}$ be a set of M points which we also - for sake of thought - assume has its elements $\{S_j, j = 1, 2, ..., M\}$ located in the plane. We call S_j a sink. For each source R_i and each sink S_j we associate a cost $c(i, j)$, and we call $\{c(i, j) : i = 1, 2, ..., N,\ j = 1, 2, ..., M\}$ the cost matrix.

We also assume that $N \leq M$.

By a *matching* we mean that we to each source in a given subset of all the sources R_i have assigned a sink S_j in such a way that no sink is assigned to more than one source. We denote the sink which is assigned to the source R_i by $S_{j(i)}$. If the subset under consideration is the whole set of sources we call the matching a *complete matching* or simply an *assignment*. We denote a matching by

$$P = \{(i(k), j(i(k)),\ k = 1, 2, ..., n\}$$

where n is the size of the subset, $i(k)$ denotes the index of a source, and $j(i(k))$ denotes the index of the sink associated to the source $R_{i(k)}$.

To every matching P we can associate a cost $c(P)$ by

$$c(P) = \sum_{k=1}^{n} c(i(k), j(i(k))$$

The assignment problem is to find a complete matching P^* such that

$$c(P^*) = \min\{c(P)\},$$

where the minimum is taken over all complete matchings.

2 The linear programming formulation.

The assignment problem can be formulated as a linear programming problem as follows. Let X be a set of $N \times M$ matrices, and let $x = \{x(i,j), \quad i = 1,2,...,N, \quad j = 1,2,...,M\}$ denote a generic matrix in X. A matrix x belongs to X if the following properties holds:

a) $x(i,j) = 0$ or 1, $i = 1,2,...N, \quad j = 1,2,...,M,$

b) $\sum_{j=1}^{M} x(i,j) \leq 1$ for $1 \leq i \leq N$;

c) $\sum_{i=1}^{N} x(i,j) \leq 1$ for $1 \leq j \leq M$.

Let $\tilde{X}$ be the subset of X for which

b') $\sum_{j=1}^{M} x(i,j) = 1$ for $1 \leq i \leq N$.

Define $L(x)$ by

$$L(x) = \sum_{i=1}^{N} \sum_{j=1}^{M} x(i,j)c(i,j).$$

Having introduced these notations we can formulate the assignment problem simply as follows:

Problem I: Find $\min\{L(x) : x \in \tilde{X}\}$.

Let us now introduce the set X^*, a larger set than $\tilde{X}$, by replacing the property in a) by the following property:

a') $0 \leq x(i,j) \leq 1$.

The linear programming formulation of the assignment problem is as follows:

Problem II. Find $\min\{L(x) : x \in X^*\}$.

Since $L(x)$ is a linear function considered as a function of the elements $\{x(i,j)\}$ of x, a minimum value of the second problem can always be found at a point where the elements of x are 0 or 1, and hence, by solving Problem II and looking for an integer solution, we will also find a solution to ProblemI.

This thus implies, that the assignment problem can be formulated as a linear programming problem, and therefore - in principal - computer programs for solving linear programming problems can be applied to the assignment problem. However standard programs for linear programming problems do not always perform well when applied to the assignment problem.

In the primal-dual algorithm we will only consider matrices x belonging to the set X; that is, we will always assume that an element of a $N \times M$ matrix x is 0 or 1.

3 The dual formulation.

In order to present the dual formulation of the assignment problem we shall first introduce the so called dual variables. Thus to each source R_i we associate a dual variable which we denote by $\alpha(i)$ and to each sink S_j we associate a dual variable which we denote by $\beta(j)$. We shall say that $\{\alpha(i),\ i = 1, 2, ..., N\}$ and $\{\beta(j),\ j = 1, 2, ..., M\}$ are *feasible* dual variables if

$$\alpha(i) + \beta(j) \leq c(i, j), \quad i = 1, 2, ..., N, \quad j = 1, 2, ..., M.$$

We shall first describe the primal-dual algorithm for the case when the number of sources and sinks are equal, that is when $N = M$. The case $N < M$ can, by enlarging the set of sources, be transferred into the case when $N = M$. Therefore until we specify something else, from now on we assume that $N = M$. We shall however still use both letters N and M.

In case $N = M$ then the set $\tilde{X}$ is such that

$$\sum_{j=1}^{M} x(i, j) = 1, \quad i = 1, 2, ..., N \tag{1}$$

and

$$\sum_{i=1}^{N} x(i,j) = 1, \quad j = 1, 2, ..., M. \tag{2}$$

Next let $x \in \tilde{X}$ and let us rewrite the expression for $L(x)$ as follows:

$$L(x) = \sum_{i=1}^{N}\sum_{j=1}^{M} x(i,j)c(i,j) = \sum_{i=1}^{N}\sum_{j=1}^{M} x(i,j)((c(i,j) - \alpha(i) - \beta(j)) + \alpha(i) + \beta(j))$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{M} x(i,j)(c(i,j) - \alpha(i) - \beta(j)) + \sum_{i=1}^{N}\sum_{j=1}^{M} x(i,j)\alpha(i) + \sum_{i=1}^{N}\sum_{j=1}^{M} x(i,j)\beta(j))$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{M} x(i,j)(c(i,j) - \alpha(i) - \beta(j)) + \sum_{i=1}^{N} \alpha(i) + \sum_{j=1}^{M} \beta(j) \tag{3}$$

where the last equality follows from the relations (1) and (2) above.

Now if we also assume that the dual variables are feasible, then the first term on the last line in the expression (3) above must be ≥ 0, and therefore it follows that

$$L(x) \geq \sum_{i=1}^{N} \alpha(i) + \sum_{j=1}^{M} \beta(j)$$

if the dual variables are feasible.

From this follows, that if we can find a matrix $x \in \tilde{X}$ and dual feasible variables $\{\alpha(i),\ i = 1, 2, ..., N\}$ and $\{\beta(j),\ j = 1, 2, ..., M\}$ such that

$$L(x) = \sum_{i=1}^{N} \alpha(i) + \sum_{j=1}^{M} \beta(j)$$

then we have found an optimal solution to the assignment problem. This is the key obsevation behind the primal-dual algorithm.

Let us write

$$D(\alpha, \beta) = \sum_{i=1}^{N} \alpha(i) + \sum_{j=1}^{M} \beta(j).$$

Thus, what we want to find is a matrix $x \in \tilde{X}$ and dual variables $\{\alpha(i),\ i = 1, 2, ..., N\}$ and $\{\beta(j),\ j = 1, 2, ..., M\}$ which are feasible, such that

$$L(x) = D(\alpha, \beta). \tag{4}$$

Again looking at the equality (3) we find that this will be accomplished if the following relations hold between the matrix x in $\tilde{X}$ and the dual feasible variables $\{\alpha(i),\ i = 1, 2, ..., N\}$ and $\{\beta(j),\ j = 1, 2, ..., M\}$:

$$c(i,j) > \alpha(i) + \beta(j) \Rightarrow x(i,j) = 0. \tag{5}$$

The strategy for the primal-dual algorithm is to find a set of feasible dual variables and a matrix x in $\tilde{X}$ such that (5) holds. Once these are found the problem is solved!

4 The primal-dual algorithm. An outline.

In this section we shall give an outline of the primal-dual algorithm.

First we need some other notions and notations. Suppose that we have a set of feasible dual variables $\{\alpha(i),\ i = 1, 2, ..., N\}$ and $\{\beta(j),\ j = 1, 2, ..., M\}$. A pair (i, j), where i is an index of a source and j is an index of a sink, such that

$$c(i,j) = \alpha(i) + \beta(j) \tag{6}$$

is called an *admissable arc*. The set of all admissable arcs will be denoted A or, if we want to emphazise the dependence of the dual variables, by $A(\alpha, \beta)$.

Now given a set A of admissable arcs, we associate a subset $X(A)$ of the set X introduced above in such a way that:

$x \in X(A)$ if $x \in X$ and $x(i,j) = 0$ if $(i,j) \notin A$.

Another notation which will be practical is the notation $n(x)$, by which we mean the number of non-zero elements in the matrix x, that is

$$n(x) = \sum_i \sum_j x(i,j).$$

Note that if the matrix $x \in X$ and also $n(x) = N$ then $x \in \tilde{X}$ and that if $x \in \tilde{X}$ then $n(x) = N$.

Let us also use the following terminology. We say that a source R_i is *full*, if

$$\sum_{j=1}^{M} x(i,j) = 1,$$

otherwise we say that the source R_i is *empty*. Similarly we say that a sink S_j is *full* if

$$\sum_{i=1}^{N} x(i,j) = 1,$$

otherwise the sink S_j is *empty*. Finally we shall call a matrix $x \in X$ a *flow*, and shall call $n(x)$ the *size* of the flow.

Having introduced these notations and notions, we note that the assignment problem is solved if we can find a set of feasible dual variables $\{\alpha(i),\ i = 1,2,...,N\}$ and $\{\beta(j),\ j = 1,2,...,M\}$ and a matrix x such that $n(x) = N$ and $x \in X(A(\alpha,\beta))$ where $A(\alpha,\beta)$ is the set of admissable arcs associated to the set of dual variables $\{\alpha(i),\ i = 1,2,...,N\}$ and $\{\beta(j),\ j = 1,2,...,M\}$.

The primal-dual algorithm will use three basic sub-algorithms which we call the *labelling routine*, the *flow change routine*, and the *dual variable change routine.*

The main purpose of the labelling routine is to connect an empty source with an empty sink. If the labelling routine ends with such a connection we say that the labelling routine ends with a *breakthrough.* Otherwise we say that the routine ends with a *non-breaktrough.*

The purpose of the flow change routine is to change the present flow in such a way that the new flow x', say, will be such that 1) $x' \in X(A)$ and 2) $n(x') = n(x) + 1$.

The purpose of the dual variable change routine is to change the present set of dual variables $\{\alpha(i),\ i = 1,2,...,N\}$ and $\{\beta(j),\ j = 1,2,...,M\}$, to a new set $\{\alpha(i)',\ i = 1,2,...,N\}$ and $\{\beta(j)',\ j = 1,2,...,M\}$, say, in such a way that the new set of admissable arcs $A(\alpha', \beta')$ will be such that the present flow x which we know belongs to $X(A(\alpha, \beta))$ also will belong to the set $X(A(\alpha', \beta'))$.

A brief description of the primal-dual algorithm is as follows.

Step 0. Determine an initial set of feasible dual variables, determine the set A of admissable arcs associated to these dual variables, and determine an initial flow $x \in X(A)$.

Step 1. Check whether $n(x) = N$. If yes we are ready. If not then:

Step 2. Start the labelling routine.

Step 3. If the labelling routine ends with a breakthrough then go to step 4. Otherwise go to step 5.

Step 4. Use the flow change routine to update the flow $x \in X(A)$ and check whether $n(x) = N$ now. If so, we are ready. Otherwise go to Step 2.

Step 5. Use the dual variable change routine to change the dual variables.

Step 6. Find the new set of admissable arcs.

Step 7. *Continue* the labelling routine. Then go to step 3.

This is it. That the primal-dual algorithm will lead to a solution follows from the properties we already have mentioned regarding the labelling routine, the flow change routine and the dual variable change routine, together with the following fact: When we continue the labelling routine after we have found the new set of admissable arcs $A(\alpha', \beta')$, at least one more sink will be labelled.

We shall now give the arguments from which it follows that the primal-dual algorithm leads to a solution of the assignment problem.

We start with some set $\{\alpha(i),\ i = 1, 2, ..., N\}$ and $\{\beta(j),\ j = 1, 2, ..., M\}$ of dual variables. We determine the set of admissable arcs A associated to this set and find an initial flow $x \in X(A)$. We can in fact always take as initial flow the flow x_0 defined by

$$x_0(i,j) = 0, \ \ i = 1, 2, ..., N, \ \ j = 1, 2, ..., M.$$

Next we check if our initial flow is in $\tilde{X}$ in which case we are ready. Otherwise we start the labelling routine (step 2). If the labelling routine ends with a breakthrough, then, by using the flow change routine (step 4), we will increase the size of the flow by one.

If instead the labelling routine ends with a non-breakthrough then we use the dual variable change routine to determine a new set of dual feasible dual variables (step 5) after which we determine a new set A' of admissable arcs associated to the new set of dual variables (step 6). Recall that one of the properties of the dual variable change routine is that the present flow will belong to $X(A')$. We shall now continue the labelling and this can always be done since the set A' contains an arc (i,j), say, such that the source R_i is labelled but the sink S_j is not yet labelled. This implies that after at most M returns to the labelling routine, the labelling routine must end in a breakthrough, and once we have a breakthrough we will use the flow change routine by which the size of the flow is increased by one.

Since, 1) the number of operations when using the labelling routine once is bounded by $C * N^2$, 2) the number of operations when using the flow change routine is bounded by $C * N$, 3) the number of operations when using the dual variable change routine is bounded by $C * N^2$, and 4) the number of operations to determine the new set of admissable arcs, after a new set of dual variables has been found, is bounded by $C * N^2$, we conclude that after a finite number of operations the primal-dual algorithm will produce a solution to the assignment problem, and that the computational complexity of the algorithm is at most $O(N^4)$.

5 A description of the labelling routine.

To complete the proof of the fact that the primal-dual algorithm solves the assignment problem in case $N = M$ we need to verify the properties regarding the labelling routine, the flow change routine and the dual variable change routine we have mentioned above. In this section we shall describe the labelling routine in detail.

The labelling routine is as follows. We start by labelling all empty sources, with a label "Empty". We call these sources the first generation of labelled sources. We then label each sink S_j for which there exists an *admissable* arc (i, j) for some index i associated to a labelled source. Whenever we find a sink that can be labelled - and is not yet labelled - we label it by R_i and we call the source R_i the father of the sink. Once a sink is labelled we do not label it again. This means that the labelling may depend on the order in which we go through (check) the admissable arcs of interest. We note that for all arcs (i, j) such that both R_i and S_j are labelled then $x(i, j) = 0$. (In fact, if R_i is labelled and hence empty then $x(i, j) = 0$ for all $j = 1, 2, ..., M$.)

We call the set of labelled sinks obtained in this way, the first generation of labelled sinks. Now, if any of the labelled sinks is empty we have a breakthrough, and we stop the labelling routine. Otherwise we continue the labelling routine as follows. For each labelled sink S_j we check if there exists a source R_i which is not yet labelled and for which $x(i, j) = 1$. If this is the case we label the source by the sink S_j, which we call the father of the labelled source. If we can not find any new source to label, we stop the labelling routine and the labelling routine stops with a non-breakthrough. Otherwise we have obtained a new set of labelled sources, which we call the second generation of labelled sources.

We now look at all admissable arcs for which the first index i of an arc

(i, j) is associated to a source which belongs to the second generation of labelled sources. If we can find a sink S_j - which is not yet labelled - with an admissable arc (i, j) associated to a source R_i of the second generation of labelled sources, then we label this sink by R_i, which we call the father of the labelled sink. We call all such labelled sinks the second generation of labelled sinks.

If we cannot find another sink to label we stop the labelling routine. Otherwise we have obtained a new set of labelled sinks.

Note that if R_i is the father of some sink S_j belonging to the second generation, then, again, $x(i, j) = 0$, since if $x(i, j) = 1$ then, since we know that $x(i, j(1)) = 1$ where $S_{j(1)}$ is the father of R_i say, this would violate condition (1), defining the set X. Note also that the set of sinks belonging to the second generation, will always be the same independently of the order in which we investigate the admissable arcs. However, which source, that will be the father of a sink of the second generation, may depend on the order of investigation.

If any of the sinks in the second generation is empty we have a breakthrough and we leave the labelling routine. Otherwise for each sink in the second generation of labelled sinks, we look for a source R_i which is not yet labelled and for which $x(i, j) = 1$. If we can find such a source we label it by S_j. If we can not find any such source the labelling routine ends with a non-breakthrough, otherwise we continue the labelling.

And we continue in this way , going back and forth between labelled sources and labelled sinks, until either we obtain a breakthrough or we end in a non-breakthrough.

When we end the labelling routine we will have a set of labelled sources and labelled sinks. If a labelled source is empty the label is simply "empty".

If a labelled source is full, its label will be a symbol of a sink and this sink will be called its father. Each labelled sink is labelled by a symbol of a source, and this source is called the father of the labelled sink.

6 The flow change routine.

The flow change routine is very simple. It is used when the labelling routine ends with a breakthrough.

What we do in the flow change routine is the following. We start at a sink (the sink) which is both labelled and empty, say $S_{j(0)}$. (At least one such sink exists). By going to its father we find a source $R_{i(1)}$ say. If this source is empty we have obtained what we call a one-step connection. We then update the present flow simply be replacing the present value of $x(i(1), j(0))$, which must be $= 0$, since both $R_{i(1)}$ and $S_{j(0)}$ are empty, by the value 1. Since the arc $(i(1), j(0))$ is admissable it is clear that the updated flow also belongs to $X(A)$ and it is also clear that the flow value has increased by one.

In case the source $R_{i(1)}$ is not empty, then we go to the sink $S_{j(2)}$ say, which is the father of $R_{i(1)}$, and from $S_{j(2)}$ we then go to the source $R_{i(3)}$ say, which is the father of $S_{j(2)}$.

Again we have two possibilities. Either the source $R_{i(3)}$ is empty or full. If it is empty we have a 3-steps connection between an empty sink and an empty source. In this case we redefine the present flow as follows.

We define $x(i(1), j(0)) = 1$ which we know was $= 0$ before. We define $x(i(1), j(2)) = 0$, which we know was $= 1$ before. And we define $x(i(3), j(2)) = 1$ which we know was $= 0$ before.

Clearly the new flow will have a size which is increased by 1. That the matrix x redefined in this way still belongs to the set $X(A)$ is clear since both

$(i(3), j(2))$ and $(i(1), j(0))$ are admissable arcs - as before, and it also clear that the new matrix x obtained in this way, still satisfies conditions (1) and (2). To see this we see that $\sum_{j=1}^{M} x(i(1), j) = 1$ still, since the only two terms which are changed in this sum are the terms $x(i(1), j(0))$ and $x(i(1), j(2))$. The former term is increased by one and the second is decreased by one and therefore the sum as before is equal to 1. By the same kind of observation we note that $\sum_{i=1}^{N} x(i, j(2)) = 1$ as before. And for the sink $S_{j(0)}$ and the source $R_{i(3)}$ it is clear that $\sum_{i=1}^{N} x(i, j(0)) = 1$ and $\sum_{j=1}^{M} x(i(3), j) = 1$ respectively.

In case $R_{i(3)}$ is not empty i.e full, then we connect $R_{i(3)}$ with its father, say the sink $S_{j(4)}$, after which we connect the sink $S_{j(4)}$, with its father say the source $R_{i(5)}$.

And in this way we continue until a father of a sink, which we have found in this way turns out to be an empty source.

Say that the source $R_{i(2n+1)}$ obtained in this way turns out to be empty. Then we say that we have found a $2n+1$-steps connection between the empty sink $S_{j(0)}$ and the empty source $R_{i(2n+1)}$. We then update the flow as follows. We redefine $x(i(2k+1), j(2k)) = 1$ for $k = 0, 1, 2, ..., n$, all of which we know were $= 0$ before, and we redefine $x(i(2k+1), j(2k+2)) = 0$ for $k = 0, 1, 2, ..., n-1$, all of which we know were $= 1$ before. Since the number of changes from a value $= 0$ to a value equal to 1 is one more than the number of changes from the value 1 to the value 0, it is clear that the size of the updated flow is increased by 1 by the flow change routine. That the new flow still belongs to the set $X(A)$ is proved in the same way as we proved it in the case we had a 3-steps connection. We skip the details in the the general case.

We hope we now have convinced the reader that the flow change routine increases the flow by 1 and that the new flow still belongs to the set $X(A)$, which was the property we needed in our argument to prove that the primal-dual algorithm always leads to a solution of the assignment problem. It should also be clear that the number of operations needed to change the flow is bounded by a constant times N.

7 The details of the dual variables change routine.

In case the labelling routine ends with a non-breakthrough we go to the dual variables change routine.

This routine starts by computing a quantity which we denote by δ. In order to define δ let L_1 be the subset of $\{1, 2, ..., N\}$ such that R_i is labelled if $i \in L_1$, let U_1 denote the subset of $\{1, 2, ..., N\}$ such that R_i is not labelled if $i \in U_1$, let L_2 denote the subset of $\{1, 2, ..., M\}$ such that S_j is labelled if $j \in L_2$, and let U_2 denote the subset of $\{1, 2, ..., M\}$ such that S_j is not labelled if $j \in U_2$.

The number δ is now defined as follows:

$$\delta = \min\{c(i,j) - \alpha(i) - \beta(j) : i \in L_1,\ j \in U_2\}. \tag{7}$$

Since the dual variables are feasible it is clear that $\delta \geq 0$. That δ can not be equal to zero is easily proved. For assume that $\delta = 0$. This implies that $c(i(0), j(0)) = \alpha(i(0)) + \beta(j(0))$ for some arc (i_0, j_0) and hence $(i(0), j(0))$ is an admissable arc, where $i(0)$ and $j(0)$ are such that $R_{i(0)}$ is labelled whereas $S_{j(0)}$ is not labelled. But the labelling process is such that the labelling goes on until a breakthrough occurs, or until no further sink or source can be labelled. But if $(i(0), j(0))$ is an admissable arc, and the source $R_{i(0)}$ is labelled then $S_{j(0)}$ would also be labelled which contradicts the fact that

$j(0) \in U_2$. Hence $\delta > 0$.

We now use δ in order to change the dual variables as follows:

$$\alpha_{new}(i) = \alpha(i) + \delta, \ \ if \ \ i \in L_1$$

$$\alpha_{new}(i) = \alpha(i) \ \ if \ \ i \in U_1$$

$$\beta_{new}(j) = \beta(j) - \delta, \ \ if \ \ i \in L_2$$

$$\beta_{new}(j) = \beta(j) \ \ if \ \ i \in U_2.$$

Let us first check that this set of dual variables also is feasible. The only way that the difference

$$c(i,j) - (\alpha(i) + \beta(j))$$

can decrease is if the source R_i is labelled and the sink S_j is unlabelled. But in this case the difference is at least δ and therefore the new set of dual variables will also be feasible.

Next let us show that the present flow x also belongs to the set $X(A')$ where A' is the set of admissable arcs which belong to the new set of dual variables. What we want to verify is that if $c(i,j) - \alpha(i) - \beta(j) > 0$ then $x(i,j) = 0$ or equivalently if $x(i,j) = 1$ then $c(i,j) - \alpha(i) - \beta(j) = 0$. Thus suppose $x(i,j) = 1$. The source R_i may either be labelled or unlabelled. If the source is unlabelled then the sink S_j must also be unlabelled since otherwise R_i would be labelled with the sink S_j as its father. Hence in this case neither $\alpha(i)$ nor $\beta(j)$ are changed and since we know that $c(i,j) - \alpha(i) - \beta(j) = 0$ before since $x \in X(A)$ we conclude that (i,j) must also be admissable after the updating in this case. If instead the source R_i is labelled, then since $x(i,j) = 1$ it is not empty, and therefore it must have a father and this father must be precisely the sink S_j. Hence again we find that $c(i,j) - \alpha_{new}(i) - \beta_{new}(j) = c(i,j) - \alpha(i) - \delta - \beta(j) + \delta = c(i,j) - \alpha(i) - \beta(j) = 0$ since $x \in X(A)$. Hence $x \in X(A')$.

The last thing we have to check is that if we continue the labelling where we left the labelling, then at least one more sink will be labelled. By continue the labelling we mean that we look at all labelled sources and look at all admissable arcs obtained by the new set of dual variables, and try to find a sink which is not yet labelled. That at least one more sink can be found which is not yet labelled follows immediately from the definition of δ. For let i and j be such that $\delta = c(i,j) - \alpha_{old}(i) - \beta_{old}(j)$ and such that R_i is labelled but S_j is not yet labelled. At least one such pair exists by the definition of δ. Then clearly (i,j) is not an admissable arc associated to the old set of dual variables, but it will of course be an admissable arc in the new set of dual variables, since $\alpha_{new}(i) = \alpha_{old}(i) + \delta$ and $\beta_{new}(j) = \beta_{old}(j)$.

Since the source R_i is labelled we can continue the labelling routine by labelling the sink S_j with the source R_i. If the sink S_j is empty we have obtained a breakthrough and we go to the flow change routine. Otherwise we can either continue the labelling by finding a source $R_{i'}$ such that $x(i',j)$=1 or we go to the dual variable change routine again.

The arguments above show that whenever we leave the dual variables change routine and go back to the labelling routine we will increase the number of labelled sinks by at least one.

It should also be clear that the number of operations needed to find the new set of dual variables and the new set of admissable arcs is bounded by $C * N^2$.

And thereby the proof of the fact that the primal-dual algorithm produces a solution to the assignment problem - in a finite number of operations - is completed.

8 The case $N < M$.

In the algorithm above we have assumed that the number of sources N is equal to the number of sinks M. To handle the case when $N < M$ we simply argue as follows.

We introduce $M - N$ extra sources $R_i,\ \ i = N+1, N+2, ..., M$ with the property that

$$c(i,j) = 0, \quad i = N+1,\ N+2, ..., M,\ j = 1, 2, ..., M.$$

We then solve the assignment problem associated to this enlarged cost matrix by using the primal dual algorithm described above. Finally by considering the matching

$$P = \{(i, j(i)),\ \ i = 1, 2, ..., N\}$$

we obtain an optimal matching for the original problem. For if there would exist another matching

$$\tilde{P} = \{(i, \tilde{j}(i)), i = 1, 2, ..., N\}$$

with a total cost satisfying $c(\tilde{P}) < c(P)$ then this matching can be used to obtain a better solution to the enlarged assignment problem and then P would not be part of an optimal matching of the enlarged problem.

Thus by enlarging the cost matrix with a number of rows whose elements are 0 we can easily solve the general assignment problem.

I want to point out that in case $N << M$ then the cost matrix will be enlarged rather much, thereby implying that the computation time will also be increased rather much compared to what a more direct method might need, such as the simplex method. However I do not know if there are smart methods when N is much smaller than M.

9 Initial updating.

We have above presented the arguments by which it follows that the primal-dual algorithm leads to the solution of the assignment problem. In this section we shall show how one can obtain a rather good initial set of feasible dual variables. However, the time gained, in comparison to taking, as initial sets of dual variables, the sets defined by $\{\alpha(i) = 0, i = 1, 2, ..., N\}$ and $\{\beta(j) = 0, j = 1, 2, ..., M\}$ is not so large.

Anyhow, as initial set of dual variables we propose the following. We first define

$$\alpha(i) = \min\{c(i, j) : j = 1, 2, ..., M\}$$

for $i = 1, 2, ..., N$. We then define

$$\beta(j) = \min\{c(i, j) - \alpha(i) : i = 1, 2, ..., N\}$$

for $j = 1, 2, ..., M$.
That these dual variables are feasible is fairly obvious from the definition.

Having defined this set of feasible dual variables, we note that for each source R_i and each sink S_j there exists at least one admissable arc. We now simply define our initial flow x' iteratively as follows. Let $S_{j(1)}$ be a sink sich that $(1, j(1))$ is an admissable arc. Set $x(1, j(1)) = 1$ and $x(1, j) = 0$ for other values of j. Let $j(2)$ be an index $\neq j(1)$ such that $(2, j(2))$ is admissable if such an index $j(2)$ exists.

If it does not exist put $x(2, j) = 0$ for $j = 1, 2, ..., M$. If it does exist put $x(2, j(2)) = 1$ and $x(2, j) = 0$ for all other indices j.

And so on.

10 Computation times.

We have implemented our algorithm in Matlab environment. The cost matrix we have chosen in our experiments is purely random in the sense that the cost between a source and a sink is simply uniformly distributed, and all costs aru independent.

Roughly speaking the the computation time is a few seconds until the size is at least above 40. When the size of the problem is 100 the computation time is 30 seconds when using a PC with a Pentium II processor, when the problem is of size 200 the computation time is approximately 6 minutes, and when the problem if of size 400 the compution time is approximately 90 minutes. These numbers indicate that the compution time is of order $O(N^4)$ for large N.

11 Comparison with the simplex method.

We have made some comparisons between our implementation of the primal-dual algorithm as described above and an algorithm based on the linear programming routine that exists in the Matlab library.

It turns out that already when the problem is of size 10 the algorithm based on the linear programming routine is noticably slower. At size 20 the computation time is approximately 1000 times slower, and for size 30 the algorithm based on the linear programming routine takes hours to found the solution whereas the computation time for the primal-dual algorithm is only a few seconds. We have not made any comparison with any other implementation of the simplex method then the one based on the LP-function that exists in MatLab.

12 Literature.

The primal-dual algorithm for the assignment problem is often called the Hungarian algorithm. The first publication was by H.W. Kuhn in 1955. See [3].

There are many textbooks on optimization theory which contain good descriptions of the primal-dual algorithm for the assignment problem. We shall here only mention two, namely [4] from the middle of the 1970s by K.G. Murty, and [1] by R.K. Ahuja, T.L. Magnanti and J.B. Orlin from the beginning of the 1990s.

This paper is also based on my work on the transportation problem for images, see e.g. [2].

References

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *NETWORK FLOWS, Theory, Algorithms and Applications*, Englewood Cliffs, Prentince Hall, NJ, (1993).

[2] T. Kaijser, *Computing the Kantorovich distance for images* J Math Imaging and Vision, **9** (1998), pp 173-91.

[3] H.W. Kuhn, *The Hungarian method for the assignment problem* Naval Research Logistics Quarterly, **2** (1955), pp 83-97.

[4] K. Murty, *Linear and Combinatorial Programming,* Wiley, New York, NY, (1976).